

PRACTICE MATERIAL FOR SPELLATHON

(GRADE 8)

1. Create a Multi-Choice Quiz with a Timer

Task: Design a quiz application that presents multiple-choice questions and includes a timer for each question.

Steps:

1. Use "ask" blocks to display questions and "say" blocks for options.
2. Implement a countdown timer using a variable that decreases every second.
3. Use "if" statements to check if the user's answer is correct and update the score.
4. Broadcast messages to transition to the next question.
5. Display the final score at the end, along with a message based on performance.

2. Build a Simple Physics Simulation

Task: Create a simulation that demonstrates basic physics concepts like gravity and motion.

Steps:

1. Choose a sprite to represent an object (e.g., a ball).
2. Use a variable to represent the object's vertical position and another for velocity.
3. Implement a "forever" loop to continuously update the position based on gravity.
4. Use "if" statements to check for collisions with the ground and apply a bounce effect.
5. Add controls to launch the object at an angle using keyboard inputs.

3. Design a Maze Game with Levels

Task: Develop a maze game where players navigate through multiple levels to reach an endpoint.

Steps:

1. Create multiple maze backgrounds and a player sprite.
2. Program the player to move with arrow keys, using "if" statements to prevent passing through walls.
3. Implement a scoring system based on time taken to complete each level.
4. Use "broadcast" messages to signal level completion and transition to the next maze.

5. Add obstacles or enemies that the player must avoid.

4. Create a Simple Social Media Simulation

Task: Simulate a basic social media feed where users can post updates and interact with each other.

Steps:

1. Design sprites for user avatars and a background for the feed.
2. Use "ask" blocks to allow users to input their posts.
3. Store posts in a list variable and display them using "for each" loops.
4. Allow users to "like" posts, updating a counter for each post.
5. Implement interactions using "broadcast" messages to update the feed in real time.

5. Make a Basic Music Player

Task: Develop a simple music player where users can play, pause, and switch between different tracks.

Steps:

1. Use sprites as buttons for "Play," "Pause," and "Next."
2. Import several sound files and store them in a list variable.
3. Use "when this sprite clicked" blocks to control playback of the current track.
4. Implement a variable to track the current track and use "if" statements to loop through the list.
5. Display the current track name on the screen, updating it as the user navigates.